

REMARKS

Claims 1-9 and 11-34 are currently pending in the subject application and are presently under consideration. Claims 1, 7-9, 15, 29, 30, and 32-34 have been amended as shown on pp. 2-8 of the Reply and claim 16 and 22 are cancelled.

I. Objection to Claims 13, 33, and 34

Claims 13, 33, and 34 stand objected to because of the following informalities:

As to claim 13, according to Amended Claim 1, a system wherein memory coupled to a processor.,, but claim 13, it states as ‘computer’. Claims 13 is cancelled and withdrawal of the objection is respectfully requested. As to claims 33 and 34, should state a ‘system’ claim NOT ‘method’ claim. Applicants’ representative has made appropriate amendments and therefore the rejection is moot and should be rejected. In the office action summary, the Examiner listed claim 4 as being objected to – in view of the content of the Final Office Action, applicants’ representative assumes that this statement is in error and the Examiner intended this to read claims 13, 33, and 34 and thus there is no objection to claim 4.

II. Rejection of Claims 15-23 and 32-34 Under 35 U.S.C. §101

Claims 15-23 and 32-34 stand rejected under 35 U.S.C. §101 because the claimed invention is directed to non-statutory subject matter. Claim 15 has been amended to overcome this rejection and thus it is requested that the rejection be withdrawn. Since claims 16-23 are dependent upon claim 15 which should have the rejection withdrawn, rejection of these claims has become moot and likewise should be withdrawn. Claim 32 has been amended to overcome this rejection and thus it is requested that the rejection be withdrawn. Since claims 33 and 34 are dependent upon claim 32 which should have the rejection withdrawn, rejection of these claims has become moot and likewise should be withdrawn.

III. Rejection of Claims 1-9 and 11-34 Under 35 U.S.C. §102(b)

Claims 1-9 and 11-34 stand rejected under 35 U.S.C. §102(b) as being anticipated by Herrmann (US 5,995,756).

For a prior art reference to anticipate, 35 U.S.C. §102 requires that “each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference.” *In re Robertson*, 169 F.3d 743, 745, 49 USPQ2d 1949, 1950 (Fed. Cir. 1999) (*quoting Verdegaal Bros., Inc. v. Union Oil Co.*, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987)).

Claim 1 recites “a host adaptor that merges at least one menu of the unmanaged application with at least one menu of the development environment, determines if the development environment has focus or if unmanaged application has focus, and enables at least one merged menu item and disables at least one merged menu item based upon a result of the determination...” While the Examiner contends this aspect is taught by Herrmann, applicants’ representative respectfully disagrees with this assertion.

Herrmann discloses constructing an application through use of another application. For example, Herrmann can use a component-based, rapid application development (RAD) environment to construct an application (See col. 6, lines 21-28). Items of the RAD environment are used to construct an application for another environment, specifically to design a user interface for another application. A user can ‘drag-and-drop’ components upon a form to create the user interface (See Fig. 3 and related discussion). With the claimed innovation, an unmanaged application and developmental environment are merged together such that the unmanaged application functions within the developmental environment. This differs from Herrmann in that Herrmann teaches an RAD environment that constructs an application for another program without directly interfacing or engaging the program during development.

Specifically, claim 1 recites “a host adaptor that merges at least one menu of the unmanaged application with at least one menu of the development environment...”, which is not disclosed by Herrmann. Herrmann states “The main window 361 itself comprises main menu 362, tool bar buttons 363, and component palette 364.” (col. 6, lines 33-34). In Herrmann, all menu items are part of the RAD environment (e.g., main menu 362, tool bar buttons 363, etc.). In contrast, claim 1 recites merging menu items from the development environment and the unmanaged application – thus two entities as opposed to one entity (e.g., the RAD environment). No merger of menu items occurs in Herrmann since all menu items are based in the RAD environment and thus this portion of the claim is not anticipated.

Additionally, claim 1 recites “a host adaptor that … determines if the development environment has focus or if unmanaged application has focus, and enables at least one merged menu item and disables at least one merged menu item based upon a result of the determination…” In short, claim 1 discloses determining which aspect has focus (e.g., the development environment or the unmanaged application) and based upon a result of the determination, different menu items can be enabled or disabled - this aspect is not disclosed by Herrmann.

First, Herrmann does not disclose determining focus between two different entities. The Examiner erroneously states that Herrmann teaches this determination through the language “Forms, such as form 371, are the focal point of nearly every application which one develops in the environment.” (col. 6, line 43-44). While both the claim and citation discuss focus, they are to be construed differently. In Herrmann, it is stated such that forms are an important part of building in the RAD environment (e.g., focus meaning important). However, in claim 1 focus is used to differentiate between the development environment and the unmanaged application (e.g., at a time, one has focus while another does not –focus is emphasis and not importance), thus being distinguished from Herrmann. Moreover, the cited portion states that the focal point of nearly every application is the form. This is an affirmative statement (e.g., are the focal point) and thus there is no need for a determination in Herrmann. While Herrmann does state “nearly every application”, this is still not enough to show determining as recited in claim 1. The use of the term ‘nearly’ conveys that an RAD environment can be constructed such that another entity is focus – however, there is still no determining on what has focus. There mere fact that an entity different from the form 371 can be a focal point does not teach that focus is determined.

Secondly, claim 1 recites enabling and disabling merged menu items as a result of the determination. As previously discussed, since there is no discussion in Herrmann concerning merging menu items of multiple entities, this element is also not taught. Moreover, Herrmann does not disclose enabling or disabling menu items in general, regardless of the how the menu items are used. It appears that the Examiner erroneously believes that placement of components on a form in Herrmann teaches enabling and disabling. While this may be true for the constructed application (e.g., placing ‘component A’ enables functionality of ‘component A’), this does not teach enabling and disabling merged menu items as recited in claim 1, specifically enabling and disabling based on a focus determination.

In addition, claim 1 recites “a designer framework that interfaces a development environment, the hosting component and designer framework interface such that the unmanaged application functions as a designer within the development environment...” In Herrmann, an RAD environment is used to construct an application for another program. However, this is an independent function where an environment and application are not used together. For example, in Herrmann, the RAD environment can be used to develop a program that can run in a supplemental program. However, the RAD environment does not interface the supplemental program, while in claim 1, the unmanaged application functions within the development environment (e.g., through interfacing). Since interfacing an unmanaged application to function within the development environment is not taught by Herrmann, each and every element is not taught and the rejection should be removed.

Claim 4 recites “an integration interface to facilitate integrating a third-party unmanaged application as a designer in the development environment.” In Herrmann, there is discussion of instillation of third-party components (col. 7, ln. 1-14), which the Examiner erroneously believes anticipates claim 4. However, this falls far short of reciting what is disclosed in claim 4. First, there is no disclosure of Herrmann of an integration interface. While component functionality can be used, there is no integration interface between a development environment and a third-party unmanaged application. In addition, Herrmann merely recites ‘third-party components’ – it is unknown if these components come from an unmanaged application (e.g., the third-party components can originate from another RAD developer) or if and/or how there is functionality as a designer.

Claim 6 recites “the host adaptor is application specific to facilitate the unmanaged application functioning as a designer within the development environment”. Nowhere in Herrmann is an application specific host adaptor disclosed. As stated previously regarding claim 1, since Herrmann does not disclose a host adaptor then an application specific host adaptor is also not disclosed. In addition, it is unclear how Herrmann discloses an entity being application specific as recited in claim 6.

Claim 7 recites “a document-hosting subcomponent that facilitates hosting a document that can be manipulated in the development environment.” Since the RAD environment in Herrmann functions alone, then there is no hosting of a document as recited in claim 7. In addition, claims 2-14 are dependent off claim 1 which should now be placed in a condition of

allowance and thus their rejection should also be removed. For at least the aforementioned reasons, claim 1 (and claims dependent therefrom) should have rejections removed and be placed into a condition of allowance.

Claim 15 recites similar limitations to that of claim 1 and thus aforementioned arguments made regarding claim 1 are also applied against claim 15. Additionally, claim 15 recites “a document-hosting subcomponent that facilitates hosting a document that can be manipulated in the development environment, the hosted document has at least two views, including a design view and a code view...” The Examiner erroneously contends that the code editor 381 and the object inspector window 391 anticipate the document views recited in claim 15. Herrmann states “The object inspector window 391 enables the user to easily customize the way a component appears and behaves in the application under development. The inspector 391 comprises an object selector field 392, a properties page 393, and an events page 394.” (col. 7, lines 15-17). As opposed to showing an item in a particular view (e.g., a design view as recited in claim 1), Herrmann discloses presenting metadata that relates to an item and thus not a different view of the item. For instance, property data can be disclosed and then event data – however, this merely changes information disclosed and does not modify a view of a document. Therefore, different views of the hosted document are not shown and thus each and every element of claim 15 is not taught by Herrmann. Moreover, the document can be opened within the IDE while the form is part of the development environment, not hosted as recited in the claim 15. In addition, in the non-final office action dated December 26, 2007, the Examiner states “Herrmann does not disclose a document-hosting subcomponent that facilitates hosting a document that can be manipulated in the development environment...” Applicants’ representative agrees with this statement – therefore it is unclear as to why this rejection is made since both parties agree the cited reference does not teach the document-hosting subcomponent.

Claim 15 also recites “a hosting component that interfaces to the unmanaged application such that the unmanaged application functions as a designer within the IDE.” Herrmann is silent regarding interfacing the unmanaged application and IDE such that Herrmann teaches use of the development environment independent of an unmanaged application. In addition, since there is no merger of the unmanaged application with the IDE in Herrmann, then there is no use of the unmanaged application as a designer. In Herrmann, the form 371 may be considered to function as a designer, but the form 371 is part of the RAD environment and not from a supplemental

source (e.g., such as an unmanaged application). Similar elements are recited in claims 24, 29, and 32 and thus this argument should be applied to those claims and their dependents.

Claim 18 recites “The system of claim 15 facilitates adding a control to the document and editing properties of the control.” The Examiner erroneously equates the form with the document. The document is an independent entity hosted in the IDE while the form is part of the development environment in Herrmann. For example, referring to Fig. 4 of the subject application for an example, the document can be a spreadsheet of a spreadsheet application – thus not a form used to retain component in assisting building of an application. Therefore, editing properties of a separate entity is not disclosed by Herrmann and thus this element is not taught by the cited reference.

Claim 19 recites “The system of claim 15 facilitates a merging of menus of the unmanaged application and the IDE.” Herrmann states “The main window 361 itself comprises main menu 362, tool bar buttons 363, and component palette 364.” (col. 6, lines 33-34). In Herrmann, all menu items are part of the RAD environment (e.g., main menu 362, tool bar buttons 363, etc.). In contrast, claim 19 recites merging menu items from the development environment and the unmanaged application – thus two entities as opposed to one entity (the RAD environment). No merger of menu items occurs in Herrmann since all menu items are based in the RAD environment.

Claims 24, 29 and 32 recite similar limitations to that of claim 1 and thus aforementioned arguments made regarding claim 1 are also applied against claims 24, 29, and 32. In addition, claim 24 recites receiving an unmanaged application while claims 29 and 32 recite obtaining the unmanaged application. Nowhere in Herrmann is obtaining or receiving an unmanaged application disclosed as recited in the aforementioned claims. As previously stated, Herrmann discloses building an application through an RAD application – however, an unmanaged application is never used and or interfaced. Therefore, since the RAD application does not use an unmanaged application there is never a purpose to obtaining and/or receiving an unmanaged application in Herrmann.

Claim 25 recites “hosting a document in the development environment such that the document is manipulated using native functionality of the unmanaged application and functionality of the development environment.” Since the RAD environment in Herrmann functions alone, then there is no hosting of a document as recited in claim 25. The Examiner

erroneously states that the RAD environment hosts the form 371 and thus anticipates hosting as recited in the claims (including claim 25). However, Herrmann states “As shown, the programming environment 360 comprises...a form 371...” (col. 6, ln. 29-30). Thus the form is part of the programming environment and therefore is not hosted as recited in appropriate claims (e.g., by use of the term comprises). In addition, there is no mention in Herrmann of use of native functionality of an unmanaged application as well as native functionality of the development environment in document manipulation.

Claim 26 recites “hosting a document in the development environment and exposing a code-behind file associated with the document such that contents of the file can be manipulated.” In addition to not disclosing hosting a document as previously discussed in the response, Herrmann is silent regarding a file associated with a document (even if including the form 371 as a document), exposing the file, and allowing contents of the file to be manipulated. Claim 32 recites a similar exposure element and thus the same argument should be applied.

Claim 27 recites “hosting a document in the development environment and exposing a code-behind file associated with each subdocument of the document such that contents of each file can be manipulated.” In addition to aspects recited in claim 26 which should also be applied to this claim, Herrmann is silent regarding manipulation regarding subcomponents of a document – even if the form 371 is a document and the component ‘dragged-and-dropped’ are subcomponents, there is never manipulation, just merely placement or non-placement..

Claim 29 recites “hosting a document in the development environment such that functionality of the development environment and the unmanaged application can be used to manipulate the document.” As previously discussed, Herrmann does not disclosed hosting a document in a development environment so a document can be manipulated based upon functionality of the development environment and the unmanaged application.

Claim 31 recites “providing a special page of preferences for the unmanaged application such that when integrated into the development environment, the unmanaged application behaves according to the preferences.” Since Herrmann does not interact with the unmanaged application, there is not providing of preferences. Additionally, integration does not occur in Herrmann so therefore Herrmann does not anticipate an action that functions when integration occurs. Moreover, Herrmann does not disclose modifying behavior of the unmanaged application during integration regarding preferences – the RAD environments has a program

function in a particular manner from the ‘drag-and-drop’, but this does not look at a preference page and modify behavior based upon the page.

Claim 32 recites “means for interfacing the unmanaged application to the development environment with a host adaptor that is specific to the unmanaged application such that the unmanaged application is accessible as a designer within the development environment; means for merging at least one menu of the unmanaged application with at least one menu of the development environment, determines if the development environment has focus or if unmanaged application has focus, and enables at least one merged menu item and disabled at least one merged menu item based upon a result of the determination; means for hosting a document in the development environment such that functionality of the development environment and the unmanaged application can be used to manipulate the document; and means for supplying manipulation functionality for the document.” Each and every element of this claim is not disclosed in Herrmann consistent with arguments presented *supra*. Additionally, there is no discussion in Herrmann regarding supplying manipulation functionality for the document.

Claims 13 and 16 are cancelled and therefore should have their rejections removed. For at least the aforementioned arguments, each and every element of claims 1-9 and 11, 12, 14, 15, and 17-34 (either directly or based upon dependency) are not taught by Herrmann and therefore their rejections should be removed and the claims placed into a condition of allowance.

CONCLUSION

The present application is believed to be in condition for allowance in view of the above comments and amendments. A prompt action to such end is earnestly solicited.

In the event any fees are due in connection with this document, the Commissioner is authorized to charge those fees to Deposit Account No. 50-1063 [MSFTP550US].

Should the Examiner believe a telephone interview would be helpful to expedite favorable prosecution, the Examiner is invited to contact applicants' undersigned representative at the telephone number below.

Respectfully submitted,
AMIN, TUROCY & CALVIN, LLP

/Ronald Charles Krosky/
Ronald Charles Krosky
Reg. No. 58,564

AMIN, TUROCY & CALVIN, LLP
127 Public Square
57th Floor, Key Tower
Cleveland, Ohio 44114
Telephone (216) 696-8730
Facsimile (216) 696-8731